

# Rank Fusion

## Information Retrieval Project

Davide Rigoni - Silvia Colucci - Alex Beccaro  
June 26, 2017



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

- 1 Generazione Run e Strategie di Base
  - Indicizzazione e Generazione Run
  - Progettazione
  - Problemi Riscontrati
- 2 Algoritmo ProbFuse
  - Algoritmo
  - Segmenti
  - Implementazione
- 3 Valutazione
  - Misure
  - Sulle 10 run di base
  - Rank fusion di base
  - Tra le migliori run
  - ProbFuse
  - Confronto
- 4 Conclusione
- 5 References

## Generazione Run e Strategie di Base

Punti essenziali dell'**indicizzazione**:

- scompattazione archivio tramite scripting
- installazione Terrier v4.1
- configurazione file *terrier.properties*  
[http://terrier.org/docs/v3.5/trec\\_examples.html](http://terrier.org/docs/v3.5/trec_examples.html)

Modelli scelti per la generazione delle **run**:

[http://terrier.org/docs/v4.1/configure\\_retrieval.html](http://terrier.org/docs/v4.1/configure_retrieval.html)

- |                  |                  |
|------------------|------------------|
| 1 TF_IDF         | 6 ln_expc2 (DFR) |
| 2 BM25           | 7 ln_L2 (DFR)    |
| 3 DFR_BM25 (DFR) | 8 DLH13 (DFR)    |
| 4 LGD (DFR)      | 9 IFB2 (DFR)     |
| 5 PL2 (DFR)      | 10 BB2 (DFR)     |

Il progetto **Java** è suddiviso nei seguenti **package**:

- *InputOutput*: classi per lettura e scrittura delle run e degli assessment
- *Normalize*: classi per la normalizzazione *standard* e per la *sum*
- *RunObject*: classi per rappresentare le run
- *RankFusion*: classi per la fusione

Tecniche di **rank fusion**:

- |           |                     |
|-----------|---------------------|
| ■ CombSUM | ■ CombANZ           |
| ■ CombMIN | ■ ProbFuse          |
| ■ CombMAX | ■ Variante ProbFuse |
| ■ CombMNZ |                     |

Durante lo sviluppo sono state individuate le seguenti parti critiche:

- 1** Scompattazione archivio, per quanto riguarda il nome, in quanto sembravano divisi in segmenti (es: Ft123.02Z)  
*Soluzione:* Rinominazione dei files
- 2** Ricerca informazioni riguardo il linguaggio markup di TREC  
*Soluzione:* Formato standard adoperato anche da Terrier
- 3** Indicizzazione della run con liste  
*Soluzione:* Utilizzo delle HashMap
- 4** Oggetti stringhe immutabili  
*Soluzione:* Utilizzo di un buffer

# Algoritmo ProbFuse

- 1 Suddivisione di ogni topic di ogni run in  $x$  segmenti
- 2 Per ogni segmento calcolare la probabilità che un documento sia rilevante usando la formula

$$P(d_k | m) = \frac{R_k}{R_k + NR_k}$$

- 3 Somma sulle varie run delle probabilità precedentemente calcolate con peso  $\frac{1}{k}$ , dove  $k$  è il numero del segmento in cui compare

$$s_d = \sum_{m \in M} \frac{P(d_k | m)}{k}$$

- 4 Lo score finale è il risultato della somma



Un aspetto molto importante è calibrare correttamente il numero di segmenti per topic

- 1 Influisce sulla dimensione dei segmenti e di conseguenza sui pesi di vari documenti
- 2 Maggiore il numero di segmenti, più vengono valorizzati i primi risultati delle run
- 3 Troppi segmenti possono avere effetto negativo: segmenti piccoli molto influenzati da singoli assessment

Non considerando la lunghezza variabile dei topic, inizialmente si era considerata come misura la lunghezza dei segmenti

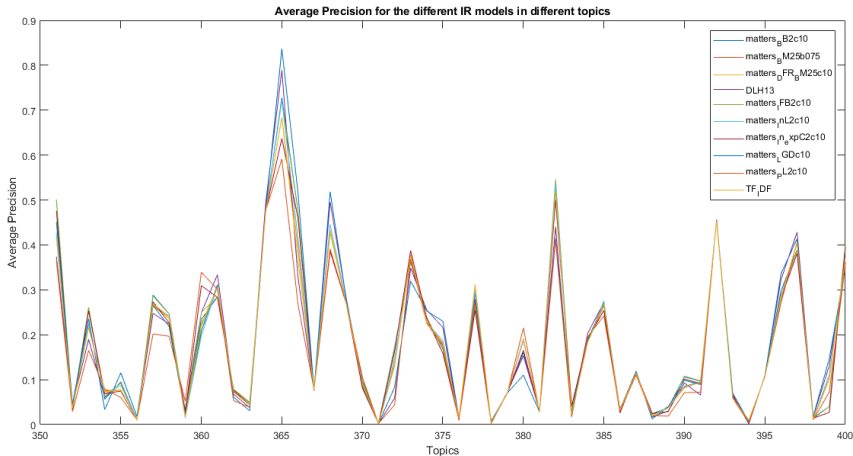
- 1 Usa stesso peso per documenti nelle stesse posizioni (assolute) invece che per documenti nella stessa porzione della run
- 2 I due approcci hanno portato a risultati diversi

# Valutazione

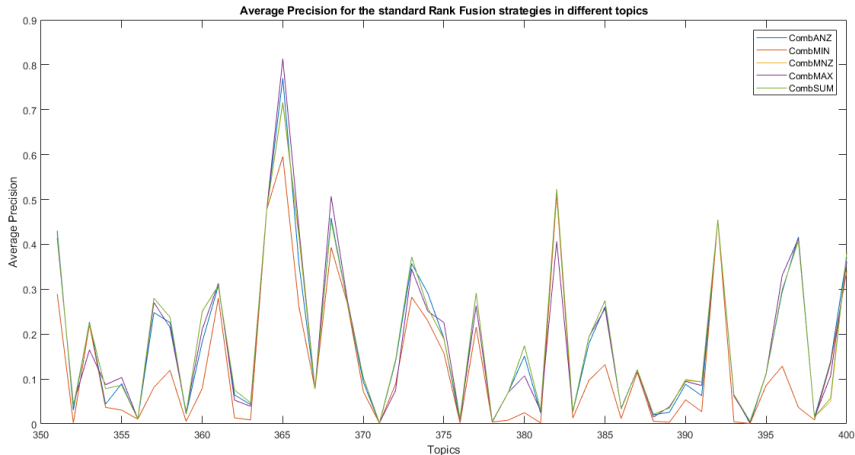
Le misure di valutazione adottate sono state le seguenti:

- *Precisione a dieci livelli di cut-off*
- *Precisione a trenta livelli di cut-off*
- *Precisione alla Recall base*
- *Average Precision*
- *Mean Average Precision (MAP)*

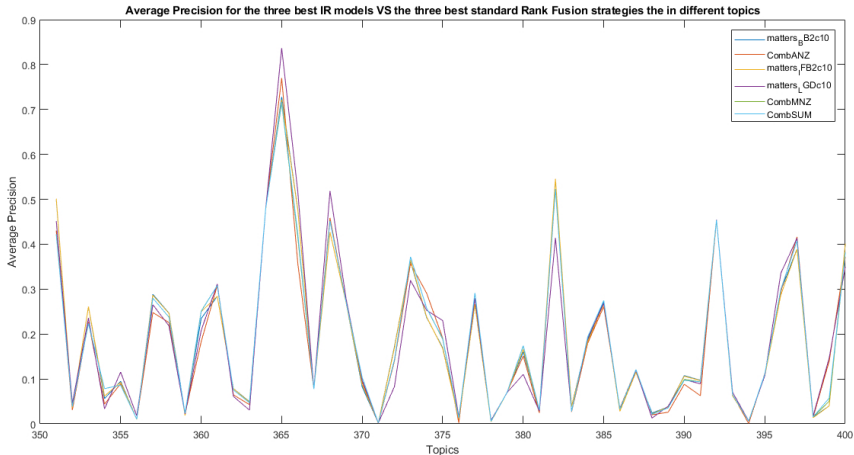
# Valutazione sulle 10 run di base

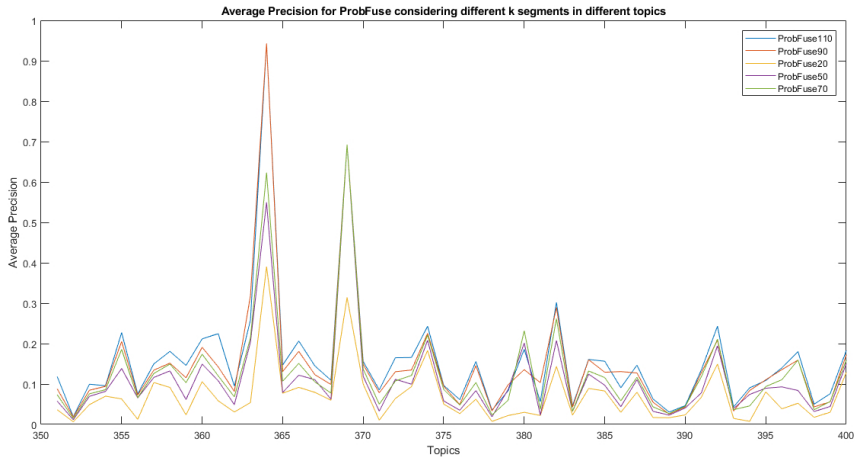


# Valutazione strategie rank fusion di base

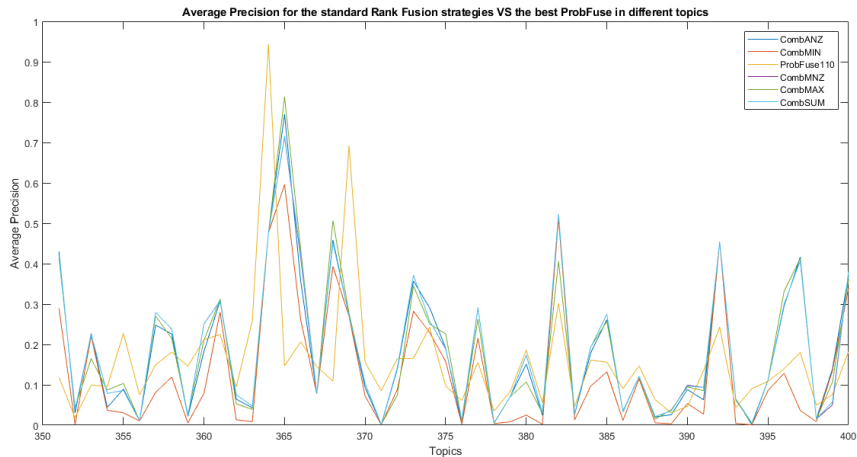


# Valutazione tra le migliori run









- 1 Indicizzazione
- 2 Modelli
- 3 Package progetto java e i problemi riscontrati
- 4 Strategie di base
- 5 ProbFuse e la sua variante
- 6 Valutazione delle 10 run di base
- 7 Valutazione strategie rank fusion di base
- 8 Valutazione ProbFuse a più k
- 9 Confronto



Fox, Edward A., and Joseph A. Shaw. "Combination of multiple searches." NIST special publication SP 243 (1994).



Lillis, David, et al. "Probfuse: a probabilistic approach to data fusion." Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006.



Montague, Mark, and Javed A. Aslam. "Relevance score normalization for metasearch." Proceedings of the tenth international conference on Information and knowledge management. ACM, 2001.

# Questions?



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

A word cloud centered around the phrase "THANK YOU". The words are in various sizes and orientations, representing different languages and expressions of gratitude. The most prominent words are "THANK", "YOU", "GRACIAS", "ARIGATO", "SHUKURIA", "JUSPAXAR", "DANKSCHEEN", "TASHAKKUR ATU", "YAQHANYELAY", "SUKSAMA", "EKHMET", "BIYAN", "SHUKRIA", "TINGKI", "BOLZIN", and "MERCİ". Other visible words include: SPASSIBO, SHACHALINYYA, HURUH, CHALTU, HUSSEJJA, HAYTEKA, RUS, VOSPAGABATIM, SHANYEBAD, ANNA, ATTO, SPASIBO, DENKAUJA, MEHCHALINYYA, MEESI, SHALGOSH, GUR, HAYUR, ENJOJU, SIKOMAO, MANTTAU, MINBAOHCHAB, PAKAAGE, KOMAPSUMNIDA, AGUYJE, EFCHARISTO, GOZAIMASHITA, MERASTAHIV, GAEJTHO, LAI, GRAZIE, MEHRBANI, PALDIES, MAAKE, and SHUKURIA.